Forecasting and Risk (BANA 4090)

Forecasting Basics (Part II)

Zhaohu(Jonathan) Fan

06/24/2021

- Main topics:
  - Evaluating forecast accuracy
  - Prediction intervals

Prerequisites

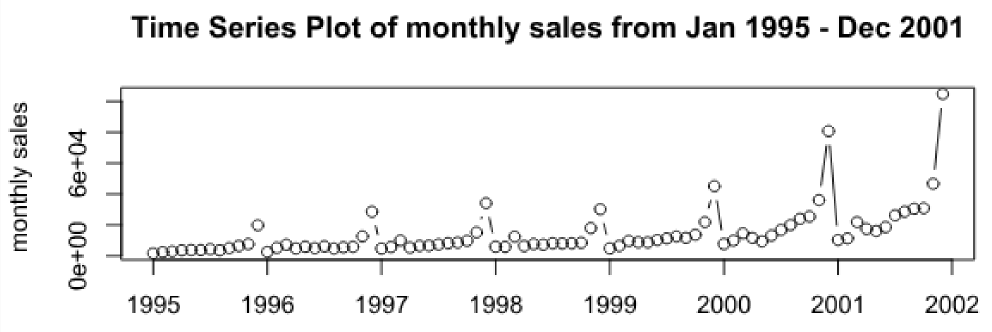
```r
# List of required (CRAN) packages
pkgs <- c(
  "ggplot2",# for drawing nicer graphics
  "fpp2",   # for using four simple forecasting models
  "forecast", #for using checkresiduals() function: a test of autocorrelation of the r
  "tidyverse",
  "readxl"

)

# Install required (CRAN) packages
for (pkg in pkgs) {
  if (!(pkg %in% installed.packages()[, "Package"])) {
    install.packages(pkg)
  }
}
```

A Toy Example

- Back in 2001, the store wanted to use the data to forecast sales for the next 12 months (year 2002).
- They hired an analyst to generate forecasts. The analyst first partitioned the data into training and validation periods, with the validation period containing the last 12 months of data (year 2001).
- She/he then fit a regression model to sales, using the training period.

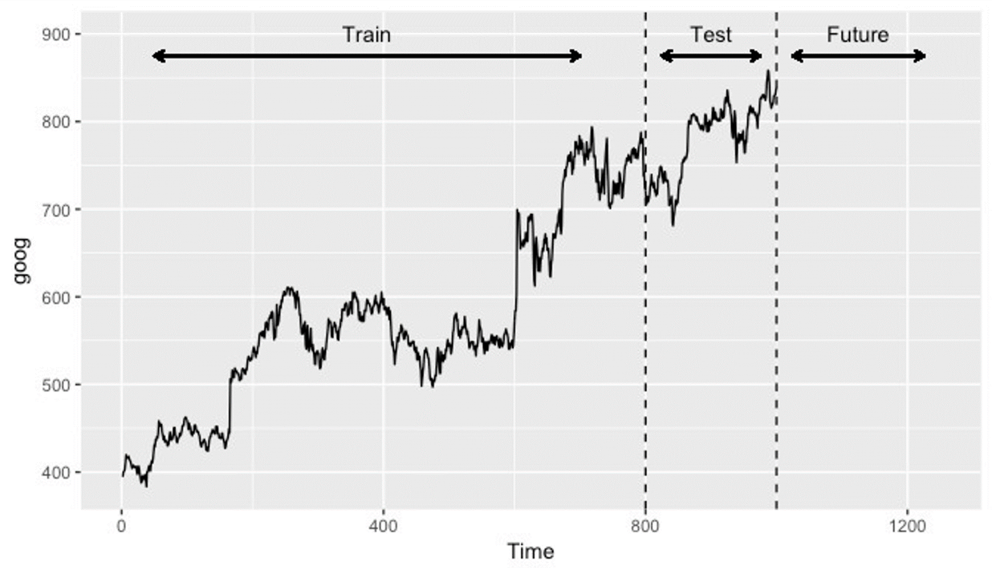


Evaluating forecast accuracy

Why Evaluate?

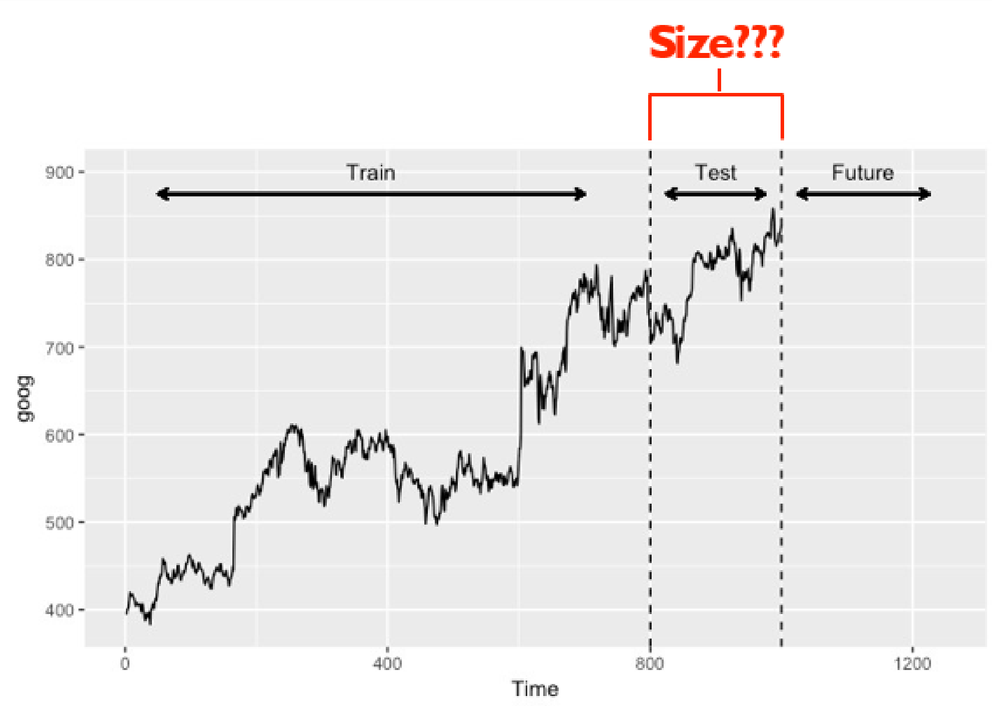
- A model which fits the data well does not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.
- Over-fitting a model to data is as bad as failing to identify the systematic pattern in the data.

Training & Test Sets



- Fit model to **training** period
- Assess performance on **test** period
- Deploy model by joining **training + validation** to forecast future.

Training & Test Sets (cont'd)

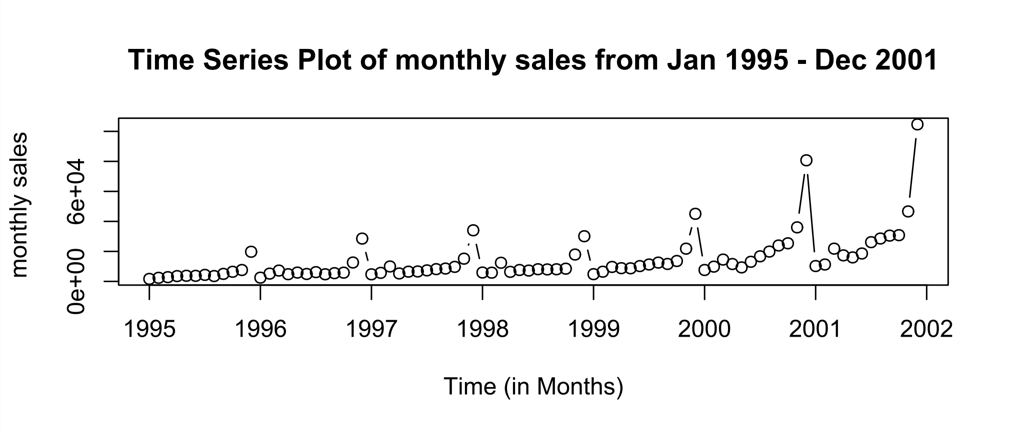


- Depends on:
  - Forecast horizon
  - Seasonality
  - Length of series

Data Example

Souvenir Sales: The file `SouvenirSales.xls` contains monthly sales for a souvenir shop at a beach resort town in Queensland, Australia, between 1995 and 2001.

```
setwd("C:/Users/fanzh/OneDrive - University of Cincinnati/UC_couse/000_Teaching_4090_S
df ← read_xlsx(path = "SouvenirSales.xlsx")
Sales_ts← ts(df$Sales, start = as.yearmon("1995-01"),end = as.yearmon("2001-12"),freq
plot(Sales_ts,type = "b",xlab= "Time (in Months)", ylab = "monthly sales ", main = "Ti
```



Data Example (cont'd)

- Q1: Why did the analyst choose a 12-month validation period?
- Q2: Partition the data in the specified manner?

Data Example (cont'd)

- Q1: Why did the analyst choose a 12-month validation period?
- A1: The forecast horizon is monthly forecasts for 1-12 months ahead. Choosing 12 months for the validation partition allows evaluating the prediction accuracy of 12-month ahead forecasts. A choice of a longer validation period might have been avoided to include recent data in the training period.

Data Example (cont'd)

- Q2: Partition the data in the specified manner?
- Data partition in R

```r
# data partition
train <- filter(df, Date < as.Date("2001-01-01"))
test  <- filter(df, Date >= as.Date("2001-01-01"))
```

Data Example (cont'd)

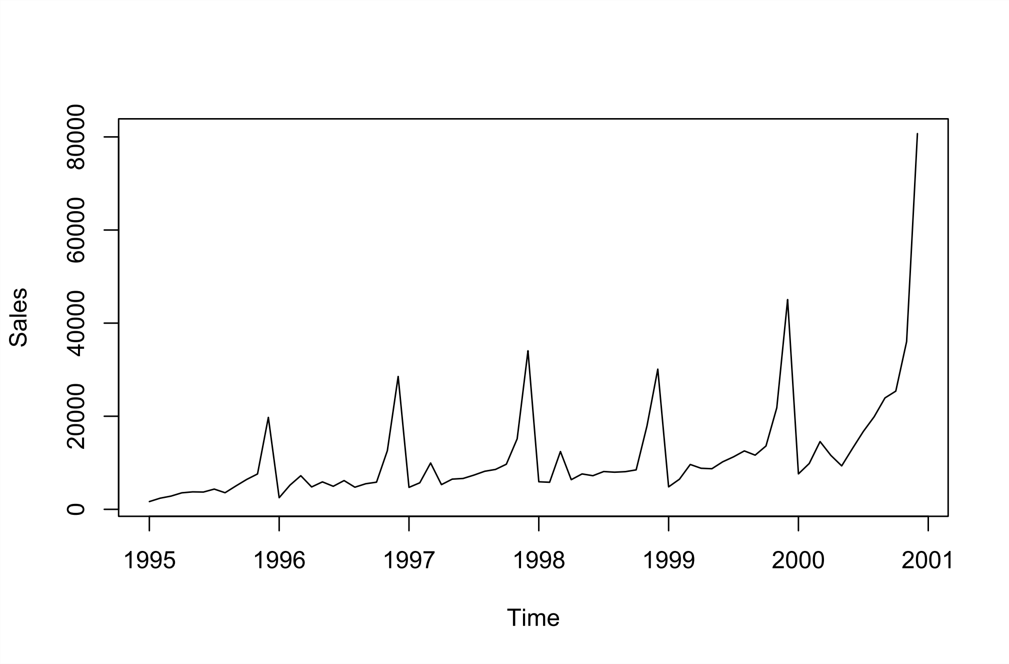
- Create time series objects

```
train.ts <- ts(train["Sales"], start = c(1995, 1), frequency = 12)
test.ts <- ts(test["Sales"], start = c(2001, 1), frequency = 12)
```

Data Example (cont'd)

- Training period

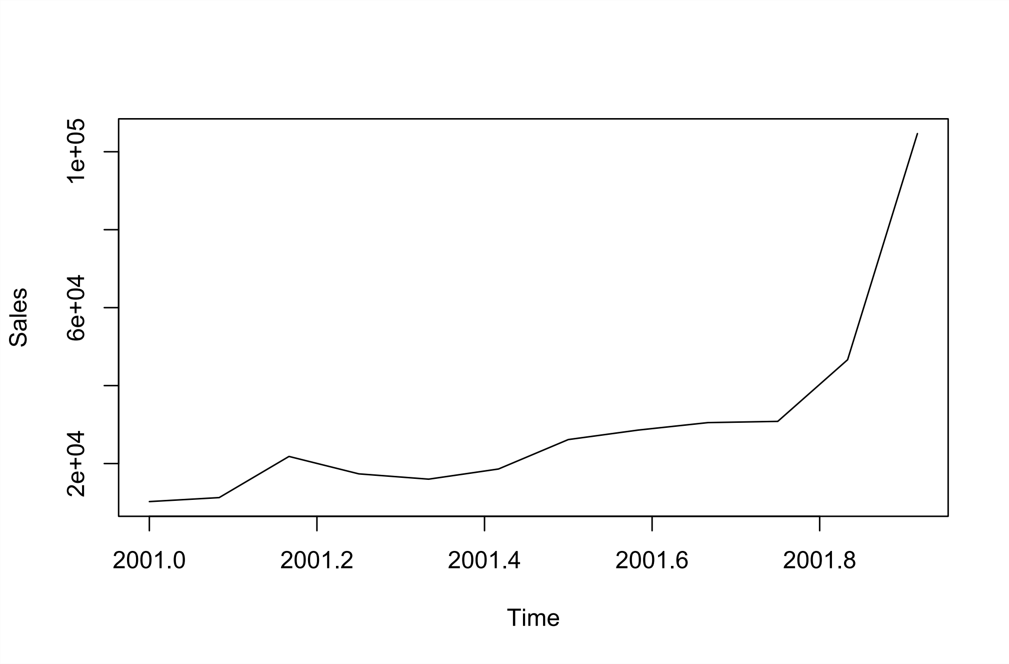
```
plot(train.ts)
```



Data Example (cont'd)

- Test period

```
plot(test.ts)
```



Naïve model

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.

```
# naive models
naive(train.ts, 12)
```

```
##          Point Forecast    Lo 80     Hi 80     Lo 95    Hi 95
## Jan 2001       80721.71 67315.75  94127.67 60219.059 101224.4
## Feb 2001       80721.71 61762.82  99680.60 51726.583 109716.8
## Mar 2001       80721.71 57501.90 103941.52 45210.077 116233.3
## Apr 2001       80721.71 53909.78 107533.64 39716.409 121727.0
## May 2001       80721.71 50745.07 110698.35 34876.389 126567.0
## Jun 2001       80721.71 47883.94 113559.48 30500.677 130942.7
## Jul 2001       80721.71 45252.87 116190.55 26476.795 134966.6
## Aug 2001       80721.71 42803.92 118639.50 22731.457 138712.0
## Sep 2001       80721.71 40503.82 120939.60 19213.758 142229.7
## Oct 2001       80721.71 38328.33 123115.09 15886.636 145556.8
## Nov 2001       80721.71 36259.16 125184.26 12722.110 148721.3
## Dec 2001       80721.71 34282.09 127161.33  9698.445 151745.0
```

Seasonal naïve model

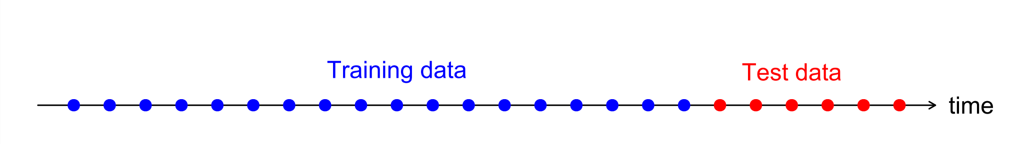
- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$, where $m =$ seasonal period and k is the integer part of $(h-1)/m$.

```
# naive models
snaive(train.ts, 12)
```

```
##          Point Forecast       Lo 80    Hi 80       Lo 95    Hi 95
## Jan 2001        7615.03   -673.8117 15903.87  -5061.6594 20291.72
## Feb 2001        9849.69   1560.8483 18138.53  -2826.9994 22526.38
## Mar 2001       14558.40   6269.5583 22847.24   1881.7106 27235.09
## Apr 2001       11587.33   3298.4883 19876.17  -1089.3594 24264.02
## May 2001        9332.56   1043.7183 17621.40  -3344.1294 22009.25
## Jun 2001       13082.09   4793.2483 21370.93    405.4006 25758.78
## Jul 2001       16732.78   8443.9383 25021.62   4056.0906 29409.47
## Aug 2001       19888.61  11599.7683 28177.45   7211.9206 32565.30
## Sep 2001       23933.38  15644.5383 32222.22  11256.6906 36610.07
## Oct 2001       25391.35  17102.5083 33680.19  12714.6606 38068.04
## Nov 2001       36024.80  27735.9583 44313.64  23348.1106 48701.49
## Dec 2001       80721.71  72432.8683 89010.55  68045.0206 93398.40
```

Training and Test Sets

- A model which fits the training data well will not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.
- Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data.
  - The test set must not be used for **any** aspect of model development or calculation of forecasts.
  - Forecast accuracy is based only on the test set.



Forecast errors

Forecast "error": the difference between an observed value and its forecast:

- $e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$,

where the training data is given by $\{y_1, \dots, y_T\}$

- Unlike residuals, forecast errors on the test set involve multi-step forecasts.
- These are *true* forecast errors as the test data is not used in computing $\hat{y}_{T+h|T}$.

Prediction intervals

Prediction intervals

- A prediction interval gives an interval within which we expect to lie with a specified probability.
- Allows us to understand the **uncertainty** that exists in a forecast.
- Should always report **point estimate** and **prediction intervals** for forecasts.
- It is common to calculate 80% intervals and 95% intervals, although any percentage may be used.
- Assuming forecast errors are normally distributed, then a 95% PI is $\hat{y}_{T+h|T} \pm 1.96\hat{\sigma}_h$.

How do we compute the prediction

Percentage	Multiplier
50	0.67
55	0.76
60	0.84
65	0.93
70	1.04
75	1.15
80	1.28
85	1.44
90	1.64
95	1.96
96	2.05
97	2.17
98	2.33
99	2.58

$F_t \pm 1.96\hat{\sigma}$

One-step prediction intervals

```
##
## Forecast method: Naive method
##
## Model Information:
## Call: naive(y = goog, h = 50)
##
## Residual sd: 8.7343
##
## Error measures:
##                     ME     RMSE      MAE        MPE      MAPE MASE       ACF1
## Training set 0.4212612 8.734286 5.829407 0.06253998 0.9741428    1 0.03871446
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1001         813.67 802.4765 824.8634 796.5511 830.7889
## 1002         813.67 797.8401 829.4999 789.4602 837.8797
## 1003         813.67 794.2824 833.0576 784.0192 843.3208
## 1004         813.67 791.2831 836.0569 779.4322 847.9078
## 1005         813.67 788.6407 838.6993 775.3910 851.9490
```

- The last value of the observed series is 813.67, so the forecast of the next value of the Google stock price is 813.67. The standard deviation of the residuals from the naïve method is 8.7343.
- Hence, a 95% prediction interval for the next value of the Google stock price is
  - 813.67 $\pm$ 1.96(8.7343) = [796.55, 830.79].
- Similarly, an 80% prediction interval is given by
  - 813.67 $\pm$ 1.28(8.7343) = [802.49, 824.85].
- Conclusion: With the naïve forecast on the next Google value, we can be 80% confident that the next value will be in the range of 802-825 and 95% confident that the the value will be between 797-831.
- Note: The value of the multiplier (1.96 or 1.28) is taken from the previous Table.

Multi-step prediction intervals

Percentage	Multiplier
50	0.67
55	0.76
60	0.84
65	0.93
70	1.04
75	1.15
80	1.28
85	1.44
90	1.64
95	1.96
96	2.05
97	2.17
98	2.33
99	2.58

- A 95% prediction interval for the h-step forecast is
  - $F_t \pm 1.96\hat{\sigma}_h, \hat{\sigma}_h = \hat{\sigma}\sqrt{h}$. where $\hat{\sigma}_h$ is an estimate of the standard deviation of the h-step forecast distribution.
- More generally, a prediction interval can be written as
  - $F_t \pm c\hat{\sigma}_h, \hat{\sigma}_h = \hat{\sigma}\sqrt{h}$. where the multiplier c depends on the coverage probability.

Multi-step prediction intervals

Example: h-step forecast=2

```
##
## Forecast method: Naive method
##
## Model Information:
## Call: naive(y = goog, h = 50)
##
## Residual sd: 8.7343
##
## Error measures:
##                     ME     RMSE      MAE        MPE      MAPE MASE       ACF1
## Training set 0.4212612 8.734286 5.829407 0.06253998 0.9741428    1 0.03871446
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1001         813.67 802.4765 824.8634 796.5511 830.7889
## 1002         813.67 797.8401 829.4999 789.4602 837.8797
## 1003         813.67 794.2824 833.0576 784.0192 843.3208
## 1004         813.67 791.2831 836.0569 779.4322 847.9078
## 1005         813.67 788.6407 838.6993 775.3910 851.9490
```

- $F_t(h)$ the forecast of the Google stock price for period 2 is 813.67.
- $\hat{\sigma}$ the standard deviation of the residuals from the naïve method is 8.7343.
- Hence, a 95% prediction interval for the next value of the Google stock price is
  - $813.67 \pm 1.96(8.7343) \times \sqrt{2}$=[789.46, 837.88].
- Similarly, an 80% prediction interval is given by
  - $813.67 \pm 1.28(8.7343) \times \sqrt{2}$ = [797.86, 829.48].